Standardizing the management and the representation of multilingual data: the MultiLingual Information Framework

Samuel Cruz-Lara, Nadia Bellalem, Julien Ducret, Isabelle Kramer

Standardizing the management and the representation of multilingual data: the MultiLingual Information Framework

Samuel Cruz-Lara, Nadia Bellalem, Julien Ducret, Isabelle Kramer

LORIA / INRIA Lorraine
Projet “Langue et Dialogue”
Campus Scientifique - BP 239
54506 Vandoeuvre-lès-Nancy
France
{Samuel.Cruz-Lara, Nadia.Bellalem, Julien.Ducret, Isabelle.Kramer}@loria.fr

Abstract

The extremely fast evolution of the technological development in the sector of Communication and Information Technologies, and in particular, in the field of natural language processing, makes particularly acute the question of standardization. The issues related to this standardization are of industrial, economic and cultural nature. This article presents a methodology of standardization, in order to harmonize the management and the representation of multilingual data. Indeed, the control of the interoperability between the industrial standards currently used for localization (XLIFF)[1], translation memory (TMX)[2], or with some recent initiatives such as the internationalization tag set (ITS)[3], constitutes a major objective for a coherent and global management of these data. MLIF (Multi Lingual Information Framework)[4] is based on a methodology of standardization resulting from the ISO (sub-committees TC37/SC3 "Computer Applications for Terminology" and SC4 "Language Resources Management"). MLIF should be considered as a unified conceptual representation of multilingual content. MLIF does not have the role to substitute or to compete with any existing standard. MLIF is being designed with the objective of providing a common conceptual model and a platform allowing interoperability among several translation and localization standards, and by extension, their committed tools. The asset of MLIF is the interoperability which allows experts to gather, under the same conceptual unit, various tools and representations related to multilingual data. In addition, MLIF will also make it possible to evaluate and to compare these multilingual resources and tools.

1. Introduction

Standards make an enormous contribution to most aspects of our lives. People are usually unaware of the role played by standards in raising levels of quality, safety, reliability, efficiency and interoperability - as well as in providing such benefits at an economical cost. The scope of research and development in localization and translation memory process development is very large; many industrial standards have been developed: TMX, XLIFF, etc. However, when we closely examine these different standards or formats by subject field, we find that they have many overlapping features. All the formats aim at being user-friendly, easy-to-learn, and at reusing existing databases or knowledge. All these formats work well in the specific field they are designed for, but they lack a synergy that would make them interoperable when using one type of information in a slightly different context. Modelization corresponds to the need to describe and compare existing interchange formats in terms of their informational coverage and the conditions of interoperability between these formats and hence the source data generated in them. One of the issues here is to explain how an uniform way of documenting such databases considering the heterogeneity of both, their formats and their descriptors.

We also seek to answer the demand for more flexibility in the definition of interchange formats so that any new project may define its own data organization without losing interoperability with existing standards or practices. Such an attempt should lead to more general principles and methods for analyzing existing multilingual databases and mapping them onto any chosen multilingual interchange format.

2. Contribution of standards

A multilingual software product should aim at supporting, for example, document indexing, automatic and/or manual computer-aided translation, information retrieval, subtitle handling for multimedia documents, etc. Dealing with multilingual data is a three steps process: production, maintenance (update, validation, correction) and consumption (use). To each one of these steps corresponds a specific user group, and a few specific scenarios. It is important to draw up a typology of the potential users and scenarios of multilingual data by considering the various points of view: production, maintenance, and consumption of these data.

The development of scenarios considers the possible limits of a multilingual product, thus the adaptations required. Normalization will also allow the emergence of new needs (e.g. addition of linguistic data like some grammatical information). Scenarios help to detect useless or superseded features which it is not necessary to implement in the standardized software application. Normalization implies a specific applicative aim, in the sense that the scenarios which should satisfy the requests must be established with precision and so being based on well “on work practices” but can envisage some possible extensions. Normalization will facilitate the dissemination (export multilingual data) as well as the integration of data (import of multilingual data from an external database).

Providing normalized multilingual products and data can be considered as an advertising for a scientific community (e.g.: consulting Eurodicautom bases on the Net). Dealing with multilingual data is an expensive process, that is why a definite application would allow a return on investment, without forgetting the promotion of

the normalization experience of your entity (industry, research center...).

3. Terminology of normalization

As "Terminological Markup Framework" [5] in terminology, MLIF will introduce a structural skeleton (metamodel) in combination with chosen data categories [6], as a means of ensuring interoperability between several multilingual applications and corpora. Each type of standard structure is described by means of a three-tiered information structure that describes:

- a metamodel, which represents a hierarchy of structural nodes which are relevant for linguistic description;
- specific information units, which can be associated with each structural node of the metamodel;
- relevant annotations, which can be used to qualify some part of the value associated with a given information unit.

3.1. What is a metamodel?

A metamodel does not describe one specific format, but acts as a kind of high level mechanism based on the following elementary notions: structure, information and methodology. The metamodel can be defined as a generic structure shared by all other formats and which decomposes the organization of a specific standard into basic components. A metamodel should be a generic mechanism for representing content within a specific context. In fact a metamodel summarizes the organization of data. The structuring elements of the metamodel are called "components" and they may be "decorated" with information units. A metamodel should also comprise a flexible specification platform for elementary units. This specification platform should be coupled to a reference set of descriptors that should be used to parameterize specific applications dealing with content.

3.2. What is a data category?

A metamodel contains several information units related to a given format, which we refer to as "Data Categories". A selection of data categories can be derived as a subset of a Data Category Registry (DCR) [6]. The DCR defines a set of data categories accepted by an ISO committee. The overall goal of the DCR is not to impose a specific set of data categories, but rather to ensure that the semantic of these data categories is well defined and understood.

A data category is the generic term that references a concept. There is one and only one identifier for a data category in a DCR. All data categories are represented by a unique set of descriptors. For example, the data category */languageIdentifier/* indicates the name of a language which is described by 2 [7] or 3 [8] digits. A Data category Selection (DCS) is needed in order to define, in combination with a metamodel, the various constraints that apply to a given domain-specific information structure or interchange format. A DCS and a metamodel can represent the organization of an individual application, the organization of a specific domain.

3.3. Methods and representation

The means to actually implement a standard is to instantiate the metamodel in combination with the chosen data categories (DCS). This includes mappings between data categories and vocabularies used to express them (e.g. as an XML element or a database field). Data category specifications are, firstly used to specify constraints on the implementation of a metamodel instantiation, and secondly to provide the necessary information for implementing filters that convert one instantiation to another. If the specification also contains styles and vocabularies for each data category, the DCS then contributes to the definition of a full XML information model which can either be made explicit through a schema representation (e.g. a W3C XML schema), or by means of filters allowing to produce a "Generic Mapping Tool" (GMT) representation.

The architecture of the metamodel, whatever the standard we want to specify, remains unchanged. What is variable are the data categories selected for a specific application. Indeed, the metamodel can be considered in an atomic way, in the sense that starting from a stable core, a multitude of data can be worked out for plural activities and needs.

4. MLIF

Linguistic structures exist in a wide variety of formats ranging from highly organized data (e.g. translation memory) to loosely structured information. The representation of multilingual data is based on the expression of multiple views representing various levels of linguistic information, usually pointing to primary data (e.g. part of speech tagging) and sometimes to one another (e.g. References, annotations). The following model identifies a class of document structures which could be used to cover a wide range of multilingual formats, and provides a framework which can be applied using XML.

All multilingual standards have a rather similar hierarchical structure but they have, for example, different terms and methods of storing metadata relevant to them. MLIF is being designed in order to provide a generic structure that can establish basic foundation for all these standards. From this high-level representation we are able to generate, for example, any specific XML-based format: we can thus ensure the interoperability between several standards and their committed applications.

4.1. Description of MLIF

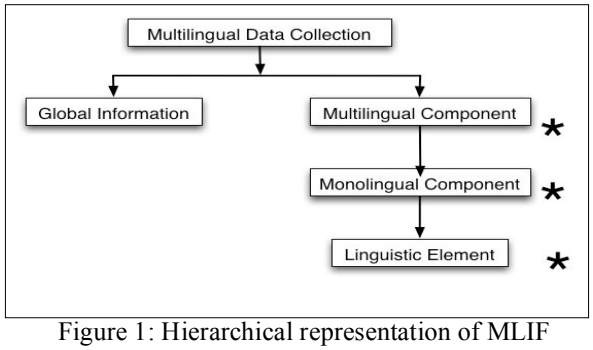

Figure 1: Hierarchical representation of MLIF

A MLIF document has a hierarchical structure as shown in Figure 1. This document will have "*Multilingual*

Data Collection" as the root level element, which content two major components: one and anly one "*Global Information*" element and one or more "*Multilingual Component*" element. The "*Global Information*" element can be considered as a header element because it contents metadata related to the document where multilingual text has been extracted and other administrative information. A "*Multilingual Component*" contains information that belongs to the linguistic unit (e.g. a single sentence or a paragraph, etc), descriptive informations (e.g. domain of application) or administrative datas (e.g. transaction, identifier, alias). Each "*Multilingual Component*" must content one or more "*Monolingual Component*" elements. A "*Monolingual Component*" is the linguistic unit in a given language. It could be a source text or a translation of this text into another language. Each of these "*Monolingual Component*" elements must contain one or more "*Linguistic Element*" elements. A "*Linguistic Element*" is the final unit of a MLIF document. It can be replaced by any metamodel such as, TMF[5], SynAF[9] or MAF[10].

For understanding what is MLIF, it is important to distinguish what depends, on the one hand, on the metamodel or, on the other hand, on the data categories. In fact, each structural node can be qualified by a group of basic or compound information units. A basic information unit describes a property that can be directly expressed by means of a data category. A compound information unit corresponds to the grouping at one level of several basic information units, which taken together, express a coherent unit of information. For instance, a compound information unit can be used to represent the fact that a transaction can be a combination of a transaction type, a responsibility, and the transaction date. Basic information units, whether they are directly attached to a structural node in the structural skeleton, or within a compound information unit, can take two non-exclusive types of values:

- an atomic value corresponding either to a simple type (in the sense of XML Schema) such as a number, string, element of a pick list, etc., or to a mixed content type in the case of annotated text;
- a reference to a structural node within the metamodel in order to express a relation between it and the current structural node.

4.2. Introduction to GMT

GMT can be considered as a XML canonical representation of the generic model. The hierarchical organization of the metamodel and the qualification of each structural level can be realized in XML by instantiating the abstract structure shown above (Figure 1) and associating information units to this structure. The metamodel can be represented by means of a generic element <struct> (for structure) which can recursively express the embedding of the various representation levels of a MLIF instance. Each structural node in the metamodel shall be identified by means of a type attribute associated with the <struct> element. The possible values of the type attribute shall be the identifiers of the levels in the metamodel (i.e., Multilingual Data Collection, Global Information, Multilingual Component, Monolingual Component, Linguistic Element).

Basic information units associated with a structural skeleton can be represented using the <feat> (for feature) element. Compound information units can be represented using the <brack> (for bracket) element, which can itself contain a <feat> element followed by any combination of <feat> elements and <brack> elements. Each information unit must be qualified with a type attribute, which shall take as its value the name of a standard data category [6] or that of a user-defined data category.

4.3. A practical example: MLIF and TMX

Now, we will use a very simple TMX example (see Figure 2) for the purpose of showing how MLIF can be mapped to other formats. As we discuss further details about MLIF, it will be clear that all features can be identified and mapped through data categories.

```
<tmx version="1.3">
    <header
        segtype="sentence"
        creationdate="19970101T163812Z"
    2   creationid="ThomasJ"
        changedate="19970314T023401Z"
        changeid="Amity">
        <note>This is an example of TMX</note>
    </header>
    <body>
        <tu tuid="0001">
            <note>Text of a note at the TU level.</note>
            <prop type="x-Domain">Translation</prop>
            <prop type="x-Project">LORIA</prop>
1           <tuv
                xml:lang="EN"
            4   creationdate="19970212T153400Z"
                creationid="BobW">
                <seg>The little cat is dead.</seg>
            3   </tuv>
            <tuv
                xml:lang="FR"
                creationdate="19970309T021145Z"
                creationid="BobW"
            4'  changedate="19970314T023401Z"
                ChangeId="ManonD">
                <seg>Le petit chat est mort.</seg>
            </tuv>
        </tu>
    </body>
</tmx>
```

Figure 2: Part of a TMX document

In Figure 2, we found structural elements of TMX : 1 represents the <tmx> root element, 2 the <header> element, 3 represents a <tu> element, 4 and 4' represent respectively the English and French <tuv> element. Next, we will match these structural elements of TMX with the metamodel of MLIF :

TMX structure	MLIF component
1 <tmx>	Multilingual Data Collection
2 <header>	Global Information
3 <tu>	Multilingual Component
4 <tuv>	Monolingual Component

Figure 3: matching TMX with MLIF components

Then, we will tag each element descriptor of TMX into 3 types: attribute, element or typed element. All these descriptors will be standardized into a MLIF descriptor element (i.e. a data category). For example the

TMX "xml:lang" attribute will be next matched with the data category named /*languageIdentifier*/ (cf figure 4).

TMX descriptor	Type	Data Categories
<note>	*element*	*/note/*
<prop type= ''x-project''>	*typed element*	*/projectSubset/*
xml:lang	*attribute*	*/languageIdentifier/*
tuid	*attribute*	*/identifier/*
<seg>	*element*	*/primaryText/*

Figure 4: typing of descriptor elements and matching with data categories.

Finally, the mapping of TMX elements into MLIF elements is represented in the following GMT file (figure 5). Note that this GMT file is nothing but a canonical representation of a MLIF document.

```
<struct type="Multilingual Data Collection">
      <struct type="Global Information">
            <brack>
                  <feat type="transaction">creation</feat>
                  <feat type="date">19970101T163812Z</feat>
                  <feat type="author">ThomasJ</feat>
            </brack>
            <brack>
                  <feat type="transaction">modification</feat>
                  <feat type="date">19970314T023401Z</feat>
                  <feat type="author">Amity</feat>
            </brack>
            <feat type="note"> This is an example of TMX</feat>
      </struct>
      <struct type="Multilingual Component">
            <feat type="identifier">0001</feat>
            <feat type="note"> It's just an example</feat>
            <feat type="subjectField">Translation</feat>
            <feat type="projectSubset">LORIA</feat>
            <struct type="Monolingual Component">
                  <feat type="languageIdentifier">EN</feat>
                  <feat type="primaryText">The little cat is dead.</feat>
                  <brack>
                        <feat type="transaction">creation</feat>
                        <feat type="date">19970212T153400Z</feat>
                        <feat type="author">BobW</feat>
                  <brack>
            </struct>
            <struct type="Monolingual Component">
                  <feat type="languageIdentifier">FR</feat>
                  <feat type="primaryText">Le petit chat est mort.</feat>
                  <brack>
                        <feat type="transaction">creation</feat>
                        <feat type="date">19970309T021145Z </feat>
                        <feat type="author">BobW</feat>
                  <brack>
                  <brack>
                        <feat type="transaction">modification</feat>
                        <feat type="date">19970314T023401Z </feat>
                        <feat type="author">ManonD</feat>
                  </brack>
            </struct>
      </struct>
</struct>
```

Figure 5: GMT representation

5. Conclusion

We have presented MLIF (MultiLingual Information Framework): a high-level model for describing multilingual data. MLIF can be used in a wide range of possible applications in the translation/localization process in several domains. This paper should be considered as a first step towards the definition of abstract structures for the description of multilingual data. The idea in a near future is to be able to implement interoperable software libraries which can be independent of the handled formats.

A first "informal" presentation of MLIF at AFNOR (Association Française pour la Normalisation - ISO's French National Body) on December 7th, 2005. We have obtained several very positive comments about our draft proposal. We are currently working on a "new work item proposal" that should be soon sent to ISO TC37 / SC4 subcommittee.

In addition, within the framework of ITEA "Passepartout" project [11], we are experimenting with some basic scenarios where MLIF is associated to XMT (eXtended MPEG-4 Textual format [12]) and to SMIL (Synchronized Multimedia Integration Language [13]). Our main objective in this project is to associate MLIF to multimedia standards (e.g. MPEG-4, MPEG-7, and SMIL) in order to be able, within multimedia products, to represent and to handle multilingual content in an efficient, rigorous and interactive manner.

6. Acknowledgements

We would like to thank Laurent ROMARY (TC37 / SC4 chairman) and Nasredine SEMMAR (CEA - LIC2M) for their useful comments and their kind help.

7. References

[1] XLIFF. (2003). XML Localisation Interchange File Format.http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff.

[2] TMX. Oscar / Lisa (2000) Translation Memory eXchange, http://www.lisa.org/tmx.

[3] ITS. W3C (2003) Internationalization Tag Set (i18n). http://www.w3.org/TR/its/

[4] S. Cruz-Lara, S. Gupta, & L. Romary (2004) *Handling Multilingual content in digital media: The Multilingual Information Framework*. EWIMT-2004 European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology. London, UK.

[5] TMF. ISO 16642 (2003) *Computer applications in terminology -- Terminological markup framework,* Genèva, International Organization for Standardization

[6] ISO 12620 (1999) : Computer applications in terminology -- Data categories,

[7] ISO 639-1 (2002) *Code for the representation of names of languages – Part 1: Alpha-2 code,* Geneva, International Organization for Standardization

[8] ISO 639-2 (1998) *Code for the representation of names of languages – Part 2: Alpha-3 code,* Geneva, International Organization for Standardization

[9] SynAF: ISO/TC37/SC4/WG2 WD 24615 Syntactical Annotation Framework.

[10] MAF: ISO/TC37/SC4/WG2 WD 24611 Morphosyntactic Annotation Framework.

[11] ITEA "Information Technology for European Advancement". Passepartout project "Exploitation of advanced AV content protocols (MPEG 4/7)" ITEA 04017.

[12] XMT. extended MPEG-4 Textual format. ISO/IEC FCD 14496-11, Information technology -- Coding of audio-visual objects -- Part 11: Scene description and application engine; ISO/IEC 14496-11/Amd 4, XMT & MPEG-J extensions.

[13] Synchronized Multimedia Integration Language (SMIL 2.0) . World Wide Web Consortium. http://www.w3.org/TR/smil20/